An Analytical Simulator for Deploying IP Telephony

K. Salah[**]   N. Darwish   M. Saleem   Y. Shaaban

Department of Information and Computer Science
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
Email: {salah,darwish,hmsaleem,yasser}@kfupm.edu.sa

Abstract. Deploying IP telephony or voice over IP (VoIP) is a major and challenging task. This paper describes an analytical design and planning simulator to assess the readiness of existing IP networks for the deployment of VoIP. The analytical simulator utilizes techniques used for network flows and queueing network analysis to compute two key performance bounds for VoIP: delay and bandwidth. The simulator is GUI-based and has an interface with drag-and-drop features to easily construct any generic network topology. The simulator has an engine that automates and implements the analytical techniques. The engine determines the number of VoIP calls that can be sustained by the constructed network while satisfying VoIP QoS requirements and leaving adequate capacity for future growth. As a case study, the paper illustrates how the simulator can be utilized to assess the readiness to deploy VoIP for a typical network of a small enterprise. We have made the analytical simulator publicly available in order to improve and ease the process of VoIP deployment.

Keywords: Network Design, Network Management, VoIP, Analysis, Simulation, Performance Evaluation

1   Introduction

When deploying a new network service such as VoIP (Voice over IP, aka IP Telephony) over existing network, network architects, managers, planners, designers, and engineers are faced with common strategic, and sometimes challenging, questions. What are the QoS requirements for VoIP? Will my existing network support VoIP and satisfy the standardized QoS requirements? If so, how many VoIP calls can the network support before upgrading prematurely any part of the existing network hardware?

These challenging questions have led to the development of a number of commercial tools to address issues related to the deployment of VoIP in data networks [1,2]. We summarize most popular commercial tools. EURESOM Jupiter II [3] has a provision to test end-to-end Quality of Service (QoS) for VoIP applications over IP networks. It considers the relationship between users' perception and network performance. NetIQ's Vivinet Assessor [4] generates RTP streams to mimic VoIP traffic between pairs of hosts and assesses the quality of these synthetic calls. BMC PATROL DashBoard [5] analyzes the impact of multimedia services on the existing network. This tool can quickly identify specific problems on the network that impact application performance. RADVISION [6] offers tightly integrated infrastructure processing components called viaIP, for desktop and meeting room conferencing. Also other companies that provide VoIP testing are Omegon [7],

---

[**] Corresponding Author: Prof. K. Salah, PO Box 5066, ICS Department, KFUPM, Dhahran 31261, Saudi Arabia

Lucent VitalSuite[8], and ViDeNet [9]. "H.323 Beacon" tool [10] is a open-source tool for assessing performance of multimedia applications including VoIP using H.323 traffic emulation.

For the most part, these tools use two common approaches in assessing the deployment of VoIP over the existing network. One approach is based on first performing network measurements and then predicting the network readiness for supporting VoIP. The prediction of the network readiness is based on assessing the health of network elements. The second approach is based on injecting real VoIP traffic into existing network and measuring the resulting delay, jitter, and loss. The second approach is sometimes referred to as network emulation.

Other than the financial cost associated with the commercial tools, none of the commercial tools gives any prediction for the total number of calls that can be supported by an existing network when considering important design and engineering factors. These factors include VoIP flow and call distribution, future growth capacity, performance thresholds, and the impact of background traffic on VoIP. In previously related work [11], an OPNET-based simulation approach was presented to determine the number of VoIP calls that can be supported for a given network topology. The paper of [12] presented detailed description of OPNET simulation models and configurations on how to generate VoIP traffic and to deal with issues such as VoIP flow, call distribution, future growth capacity and background traffic. Also in [12], an analytic approach based on the principles of queuing networks was presented to determine approximately the number of VoIP calls an existing data network can support.

The work of [11] required the availability of OPNET as well as an adequate degree of competence in using OPNET. There is a considerable learning curve required and enormous effort needed in configuring different aspects of the simulation, and more importantly in interpreting the simulation results. In addition, the simulation runs took very long time. For the case study presented (of a typical network topology of a small enterprise), one simulation run took at least 15 hours on a powerful Sun E450 machine. At least 5 simulation runs had to be performed before accurate results can be obtained. As for [12] in which an analytic approach was used, a typical network designer or engineer needs to have good understanding of queuing theory and MATLAB. One of the other major drawbacks in both approaches in [11] and [12] is not being generic. A change in network topology requires a major change and re-run of the simulation or computation. This can be very annoying and painful especially if a network engineer starts tuning the existing network topology under study. A network engineer can tune the network topology by introducing modifications (e.g., addition or placement of a switch or a router) and then examine how a small change would impact the results.

In order to quickly be able to assess network readiness for deploying multimedia service such VoIP and remedy the shortcomings of [11] and [12], a GUI-based simulator has been developed. The simulator has an interface that is GUI with the ability of building network topology by clicking and dropping (i.e., comparable to building a network in OPNET). The simulator utilizes the analytical approach introduced in [11] in a generic fashion. In other words the simulator has drag-and-drop features to construct a generic network topology and feed it into the analytical engine. The simulator also allows users to set and configure variety of settings and parameters related VoIP deployment. The analytical engine runs an optimized algorithm based on the analytical approach in [12]. The simulator gives results on how many VoIP calls can be supported within seconds. The user can easily tune the network configurations and parameters and determine the results within seconds. As a case study, we will illustrate how the simulator can be used to deploy VoIP for a typical network of a small enterprise. The results obtained by the simulator will be compared to those reported using OPNET simulation and MATLAB in [11,12]. An early attempt to develop an analytical simulator was presented in [13]. The simulator presented in [13] lacks so many of the powerful features (such as drag-and-drop, building generic network, parameter configuration, saving files, etc.) of the simulator presented in this paper.

The rest of the paper is organized as follows. Section 2 presents an analytical approach to assess the readiness of IP telephony. Section 3 describes a GUI-based analytical tool that has an engine that implements our proposed analytical approach. Section 3 also gives an example of a typical network topology of a small enterprise to be used as a case study for validating results. Finally, Section 4 concludes the study and identifies future work.

2   Analytical Approach

VoIP is bounded by two important performance metrics. First is the available bandwidth. Second is the end-to-end delay. The actual number of VoIP calls that the network can sustain and support is bounded by those two metrics. Depending on the network under study, either the available bandwidth or delay can be the key dominant factor in determining the number of calls that can be supported.

Our analytical approach considers important factors as background traffic, traffic flow, call distribution and growth factor. For background traffic, network measurements must be performed to determine the traffic rates in bps (bits per second) and pps (packets per second) for links directly connected to the router and switches. Traffic flow has to do with the path that a voice call travels through. Of course, multiple flows to carry voice calls can exist per network. Call distribution has to do with the percentage of calls to be established within and outside of a floor, building, or department. In a sense, each flow has a call distribution percentage that describes how many calls are passing through this flow or path in relation to the total calls in the network. As

for future growth factor, it depends on the projected growth in users, network services, business, etc. In our case study, we will ascertain that a default of 25% of the available network capacity is reserved for future growth and expansion. We will apply this evenly to all network resources of the router, switches, and switched-Ethernet links. However, this growth factor can be adjusted individually and per network element by the VoIP Simulator.

2.1   Bandwidth Bottleneck Analysis

In determining the bandwidth requirement and delay, we will follow the widely-adopted recommendations of H.323, G.711, and G.714 standards for VoIP QoS requirements [14,15]. Table 1 compares some commonly-used ITU-T standard codecs and the amount of one-way delay that they impose. To account for upper limits and to meet desirable quality requirement of ITU recommendation P.800 [16], we will choose G.711u codec standards for the required delay and bandwidth. G.711u yields around 4.4 MOS rating. MOS, *Mean Opinion Score*, is a commonly used VoIP performance metric given in a scale of 1 to 5, with 5 is the best. However, with little compromise to quality, it is possible to implement different ITU-T codecs that yield much less required bandwidth per call and relatively a bit higher, but acceptable, end-to-end delay. This can be accomplished by applying compression, silence suppression, packet loss concealment, queue management techniques, and encapsulating more than one voice packet into a single Ethernet frame. It is important to note that our analytical simulator (as shown later in Figure 3(a)) can be configured for any codec standard. G.711 is the default standard used by out simulator.

Table 1. Common ITU-T codecs and their defaults

Codec	Data rate (kbps)	Datagram size (ms)	A/D Conversion delay (ms)	Combined bandwidth (bi-directional) (kbps)
G.711u	64.0	20	1.0	180.80
G.711a	64.0	20	1.0	180.80
G.729	8.0	20	25.0	68.80
G.723.1 (MPMLQ)	6.3	30	67.5	47.80
G.723.1 (ACELP)	5.3	30	67.5	45.80

According to G.711u standard which we adopted as a case of illustration, the required bandwidth for a voice call in one direction is 50 pps or 90.4 kbps. G.711u codec samples 20ms of voice per packet. Therefore, 50 such packets need to be transmitted per second. Each packet contains 160 voice samples in order to give 8000 samples per second. Each packet is sent in a single Ethernet frame. With every packet of size 160 bytes, headers of additional protocol layers are added. These headers include RTP + UDP + IP + Ethernet with preamble of sizes 12 + 8 + 20 + 26, respectively. Therefore, a total of 226 bytes, or 1808 bits, needs to be transmitted 50 times per second, or 90.4 kbps, in one direction. For both directions, the required bandwidth for a single call is 100 pps or 180.8 kbps assuming a symmetric flow.

Bandwidth bottleneck analysis is an important step to identify the bottle network element, whether it is a node or a link, that sets a theoretical upper limit on the number of VoIP calls that can be supported by the network. The problem here is simpler than the maximum network flow problem [17]. The latter problem does not set a usage condition on a link other than its capacity, whereas for our situation the link (and node) usage is determined based on some given call flow distribution.

For a network that has N network elements $(1... N)$, let A_i denote the available bandwidth for network element i, which is defined as:

$$A_i = (1 - u_i)C_i,$$

where C_i is the capacity of network element i and u_i is its current utilization. The capacity C_i is the maximum possible transfer or processing rate. Therefore, the maximum number of calls that can be supported by a network element i can be expressed in terms of A_i as:

$$MaxCalls_i = \frac{A_i(1 - growth_i)}{CallBW},$$
(1)

where $growth_i$ is the growth factor of network element i, and takes a value from 0 to 1. $CallBW$ is the VoIP bandwidth for a single call imposed on i. The total bandwidth that will be introduced as a result of adding one call on link is 50pps in one direction, and another 50pps in the opposite direction. However for switches and routes, the extra bandwidth introduced per call will be 100pps.

In order to find the bottleneck network element that limits the total number of VoIP calls, one has to compute the maximum number of calls that can be supported by each network element, as in equation (1), and the percentage of VoIP traffic flow passing by this element. The percentage of VoIP traffic flow for network element i denoted as $flow_i$, can be found by examining the distribution of the calls. The total number of VoIP calls that can be supported by a network can be expressed as

$$TotalCallsSupported = \min_{i=1,...,N}\left(\frac{MaxCalls_i}{flow_i}\right).$$
(2)

2.2 Delay Analysis

In order to achieve a natural voice conversation, the end-to-end upper bound delay (sometimes termed latency) for a voice packet should be kept to minimal. Essentially, such a delay can broken into at least three contributing components, which are as follows (i) voice sampling or frame grabbing, encoding, compression,

and packetization delay at the sender (ii) propagation, transmission and queuing delay in the network and (iii) buffering, decompression, depacketization, decoding, and playback delay at the receiver. According to recommendations by ITU [3], when delays are less than 150 ms, most interactive applications, both speech and non-speech, will experience essentially transparent interactivity. For voice, the end-to-end delay is sometimes referred to by M2E or Mouth-to-Ear delay.

Therefore, we must always ascertain that the worst-case end-to-end delay for all the calls must be less than 150 ms. It should be kept in mind that our goal is to determine the network capacity for VoIP, i.e. the maximum number of calls that existing network can support while maintaining VoIP QoS. This can be done by adding calls incrementally to the network while monitoring the threshold or bound for VoIP delay. When the end-to-end delay, including network delay, becomes larger than 150 ms, the maximum number of calls can then be known.

There are three sources of delay for a VoIP stream: sender, network, and receiver. An expression is given in [4] to compute the end-to-end delay D for a VoIP flow in one direction from sender to receiver.

$$D = D_{pack} + \sum_{h \in Path}(T_h + Q_h + P_h) + D_{play} ,$$

where D_{pack} is the delay due to packetization at the source. At the source, there is also D_{enc} and $D_{process}$. D_{enc} is the encoder delay of converting A/D signal into samples. $D_{process}$ is the PC of IP phone processing that includes encapsulation . In G.711, D_{pack} and D_{enc}, are 20 ms and 1ms, respectively. Hence, it is appropriate for our analysis to have a fixed delay of 25 ms being introduced at the source, assuming worst case situation. D_{play} is the playback delay at the receiver, including jitter buffer delay. The jitter delay is at most 2 packets, i.e. 40ms. If the receiver's delay of $D_{process}$ is added, we obtain a total fixed delay of 45 ms at the receiver. $T_h + Q_h + P_h$ is the sum of delays incurred in the packet network due to transmission, queuing, and propagation going through each hop h in the path from the sender to the receiver. The propagation delay P_h is typically ignored for traffic within a LAN, but not for a WAN. For transmission delay T_h and queueing delay Q_h we apply queueing theory. Hence the delay to be introduced by the network, expressed as $\sum_{h \in Path}(T_h + Q_h)$, should not exceed (150 – 25 – 45) or 80 ms.

We utilize queueing analysis to approximate and determine the maximum number of calls that the existing network can support while maintaining a delay of less than 80ms. In order to find the network delay, we utilize the principles of Jackson theorem for analyzing queueing networks. In particular, we use the approximation method of analyzing queueing networks by decomposition discussed in [5]. In this method, the arrival rate is

assumed to be Poisson and the service times of network elements are exponentially distributed. Analysis by decomposition is summarized in first isolating the queueing network into subsystems, e.g., single queueing node. Next, analyzing each subsystem separately, considering its own network surroundings of arrivals and departures. Then, finding the average delay for each individual queueing subsystem. And finally, aggregating all the delays of queueing subsystems to find the average total end-to-end network delay. The queueing models of network elements for link, switch, and router are presented in [6].

In order to determine the maximum number of calls that can be supported by an existing network while maintaining VoIP delay constraint, we devise a comprehensive algorithm that basically determines network capacity in terms of VoIP calls. Algorithm 1 is essentially the analytical simulator's engine for computing the number of calls based on delay bound. Calls are added iteratively until the worst-case network delay of 80 ms has reached.

Algorithm 1: Compute maximum number of calls based on VoIP delay constraint

---

Input: n : number of network elements

        $\lambda[1..n]$: background traffic for network elements $1,2,..n$

        $Delay[1..n]$: delay for network elements $1,2,..n$

        P : set of call-flow paths (p) where p is a subset of $\{1,2,..n\}$

Output: MaxCalls: maxmimum number of calls

λ_{VoIP} $\leftarrow$ 100pps, or 180.8kbps;

VoIP_MaxDelay $\leftarrow$ 80; // network delay for VoIP call in ms

MaxDelay $\leftarrow$ 0;

MaxCalls $\leftarrow$ -1;

$Delay[1..n] \leftarrow$ 0;

while MaxDelay < VoIP_MaxDelay **do**

    1. MaxCalls $\leftarrow$ MaxCalls + 1

    2. Generate a call according to call distribution and let p_c be its flow
       path

    3. **for each** element i in p_c **do**

                $\lambda_i \leftarrow \lambda_i + \lambda_{VoIP}$

                if i is a link **then**

                    $Delay_i \leftarrow (1-\lambda_i/2\mu_i)/(\mu_i - \lambda_i)$

                Else

                    $Delay_i \leftarrow 1/(\mu_i - \lambda_i)$

                end if

        end for

    4. **for each** p in P where $p \cap p_c \neq \phi$ **do**

                PathDelay(p) $\leftarrow \sum_i Delay_i$, where i is a network element in path p

                if PathDelay(p) > MaxDelay **then**

                    MaxDelay $\leftarrow$ PathDelay(p)

                end if

        end for

end while

It is to be noted that in Step 2 of Algorithm 1 that a uniform random number generator is used to generate VoIP calls according to the call distribution. Call distribution must be in the form of values from 1 to 100%. Also, the delay computation for the link in Step 3 is different than other network elements such as switches and routers. For the links, it is more appropriate to use the average delay formula for *M/D/1* as the service rate μ is almost constant. However for the switches and routers, it is more appropriate to use the average delay formula for *M/M/1* as the service rate μ is variable since the routers and switches are CPU-based. For the links, the average delay per packet is calculating first using the average bit delay and then multiplying it by the packet size which is 1808 bits to get the delay per packet. For this, the link service rate and incoming rate have

to be all in bps. However for switches and routers, the calculation is all done in pps. In the algorithm above, it is worth noting that the link delay calculation in Algorithm 1 is for a unidirectional link. The total bandwidth that will be introduced as a result of adding one call on link is 50pps in one direction, and another 50pps in the opposite direction. However for switches and routes, the extra bandwidth introduced per call will be 100pps.

3   The GUI-based Analytical Simulator

In order to make use of the analytical approach presented in Section 2 and to ease the implementation of the algorithms of throughput and delay analysis for any given generic network of interest, an analytical simulator has been developed.[1]   We have made the simulator publicly available to improve and ease the process of deploying VoIP for network designers and planners.   The simulator's engine implements the analytical approach of throughput and delay analysis described in Section 2. The simulator is GUI-based with drag-and-drop feature to build any generic network topology.   In order to accomplish this, the simulator has been developed in C# programming language on top of the open source diagramming tool known as the Netron Core available at http://www.orbifold.net/netron. The classes in Netron Core are easily extendable through inheritance. Many different applications can be developed on top of the Netron Core and the code can be re-used in different contexts. For our case many features have been added particularly for building network topology. As the network topology is constructed, objects such as nodes, links, call paths, as well as call distributions are stored in hash tables. Hash tables are used for fast access as we need to lookup for nodes and links frequently in our algorithm. The simulator has the essential File, Edit, View, and drawing options. Figure 1 shows the main user interface of the VoIP tool. It has two tabbed panes: left pane and the right pane.

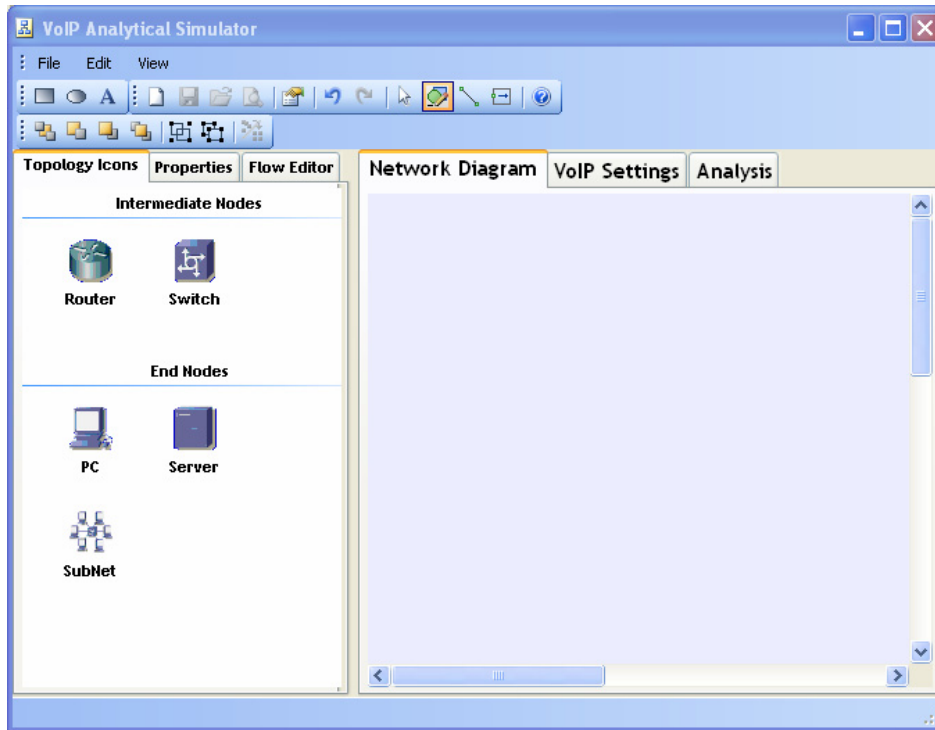[1] The simulator can be downloaded from http://www.ccse.kfupm.edu.sa/~salah/VoIP_Analytical_Simulator.rar

Figure 1. Interface of VoIP Analytical Simulator

3.1 The left pane

The left pane has three tabs, namely *Topology Icons, Properties* and *Flow Editor*, as depicted in Figure 2. Figure 1 shows the *Topology Icons* tab which has all the node icons used to build the desired network topology. The network topology icons that simulator has currently Switch, Router, PC, Server and Subnet. The network topology is built by dragging and dropping the required shape on to the stage. The stage is located on the *Network Diagram* tab of the right pane. Once the network icons are placed on the stage, they can be connected by selecting the *link* or *line* icon shown on the top main menu bar of the simulator.

Figure 2(a) shows the *Properties* tab which is used to specify the properties of the network elements of nodes, links, and subnets (such as name, bandwidth or capacity, location, width, height and the background traffic. Properties for different network elements can be selected by simply clicking on the desired network icon of the network topology. Each network element has different network properties with default values. For topology icons, users can change their sizes and locations by changing their width and height fields in their respective property window. Figure 2(a) shows the properties of a link which shows not only the capacity of the link, but also the background traffic in both Mbps and pps.

Figure 2(b) shows the *Flow Editor* tab which is used to define the VoIP flows or paths of calls in the network. A new flow can be added by clicking first "*Add New*" button and then entering a flow name. Next, to specify the path of the flow, the "*Select*" button must be clicked. Then the path details are simply inserted hop-by-hop by going to the network topology already drawn on the "*Network Diagram*" of the right pane, and hitting the network elements one-by-one. The constructed flow is bidirectional. The percentage of the total calls passing through this path is specified in the "*Call Distribution (%)*" field. The user has to click on "*Save*" when done adding the flow. For example, Figure 2(b) shows the path details and percentage of a flow named "*F1C1-VoIP Gateway*" established from a node on Floor 1 to a VoIP gateway node passing through elements of switches and routers. Once "Save" is clicked, this follow will appear in the list of "*Stored Flows*". The total call distribution of all flows must not exceed 100%.

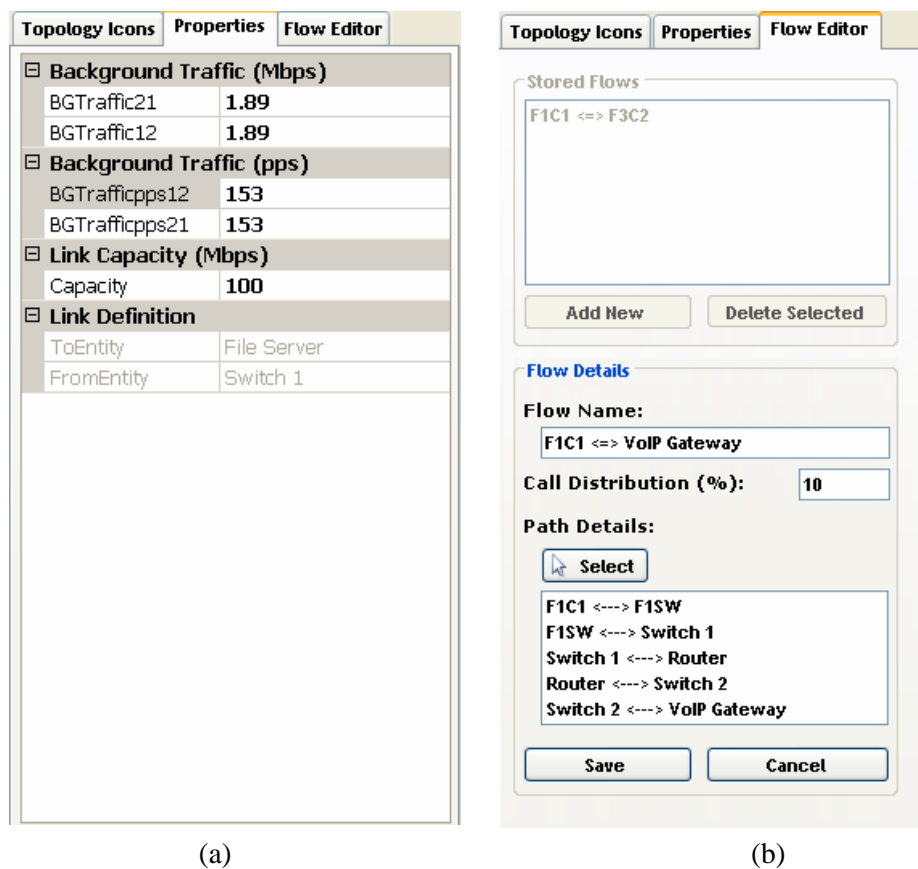

(a)   (b)

Figure 2. Properties of network elements (a) and Flow Editor of VoIP flows (b)

3.2   The right pane

The right pane also has three tabs namely, *Network Diagram, VoIP Settings* and *Analysis*. Figure 1 shows the *Network Diagram* pane which is the stage for building the topology. Any generic network topology can be built

by simply dragging an dropping networking elements of "*Topology Icons*" into the stage and connecting them through links. Links can be selected from the top main menu bar of the simulator. Figure 3(a) shows the default VoIP settings according to ITU G.711u standards, specifically for bandwidth, packet size and end-to-end latency. The users can restore or change these settings. *VoIP Settings* window allows the users to set the future growth percentage of each network element, as well as breaking the 150ms end-to-end delay into three separate delays of sender, receiver, and network. Results can be obtained through the *Analysis* tab shown in Figure 3(b). Before obtaining the results, the user can verify the models via "*Verify Model*" option to check for any errors in settings or configuration. Errors may include missing links or flows in the network topology or missing configuration. There are two different analytical reports showing the number of calls that can be supported. One report is based on the throughput analysis and the other is based on delay analysis. The details of these reports will be discussed next in Section 3.3.
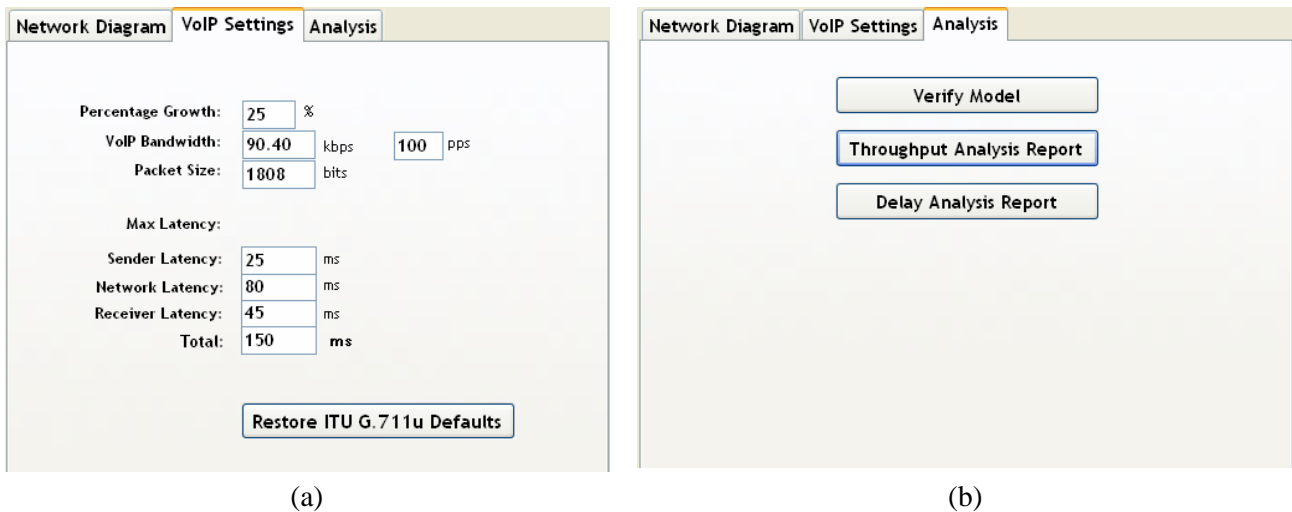

(a)   (b)

Figure 3. VoIP Settings (a) and verifications and reports (b)

3.3   Case Study

In order to illustrate the simulator's options further we give an example. Figure 4 illustrates a typical network topology of a small enterprise residing in a high-rise building. The enterprise has a network of three floors or departments. The network shows the VoIP nodes of an H.323 *gatekeeper* and *gateway*. The *gatekeeper* node handles signaling for establishing, terminating, and authorizing connections of voice calls. The VoIP *gateway* is responsible for converting VoIP calls to/from the Public Switched Telephone Network (PSTN). Other hardware requirements include a VoIP client terminal, which can be a separate VoIP device, i.e., IP phones, or a typical PC or workstation that is VoIP-enabled. This topology was also reported in [11,12] and used here for validation and comparison purposes. We will also use the same parameter values for background traffic, traffic flow, and call distribution as those reported in [11,12]. According to [18,19], the capacity C_i for the assumed

model router and switch, is 25,000pps and 1.3M pps, respectively. All links are switched full-duplex Fast (100Mbps) Ethernet.
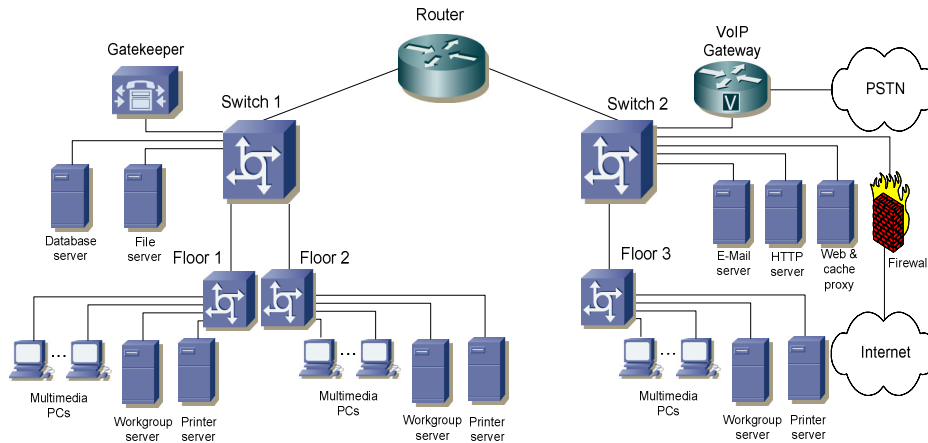

Figure 4. Network topology with VoIP Components

Figure 5 shows the corresponding network model constructed by the VoIP Simulator for the network topology of Figure 4. In order to avoid having numerous PC nodes or IP phones per floor representing end-users (and therefore clutter the network topology diagram), Floor LANs have been simply modeled as a LAN that enclose an Ethernet switch and three designated Ethernet PCs used to model the activities of the LAN users. For example, Floor 1 has three nodes (labeled as F1_C1, F1_C2, and F1_C3). F1_C1 is a source for sending voice calls. F1_C2 is a sink for receiving voice calls. F1_C3 is a sink and source of background traffic. This model allows for generating background traffic and also establishing intra-floor calls or paths from F1_C1 and F1_C2, and passing through the floor switch of F1SW. The sending and sinking PC nodes of VoIP (e.g., F1_C1 and F1_C2) have infinite capacity and there is no limit on how many calls can be added or received by them. We ignore the signaling traffic generated by the *gatekeeper*. We base our analysis and design on the worst-case scenario for VoIP call traffic. The signaling traffic involving the *gatekeeper* is only generated prior to the establishment of the voice call and when the call is finished. This traffic is relatively limited and small compared to the actual voice call traffic. In general, the *gatekeeper* generates no signaling traffic throughout the duration of the VoIP call for an already established on-going call.
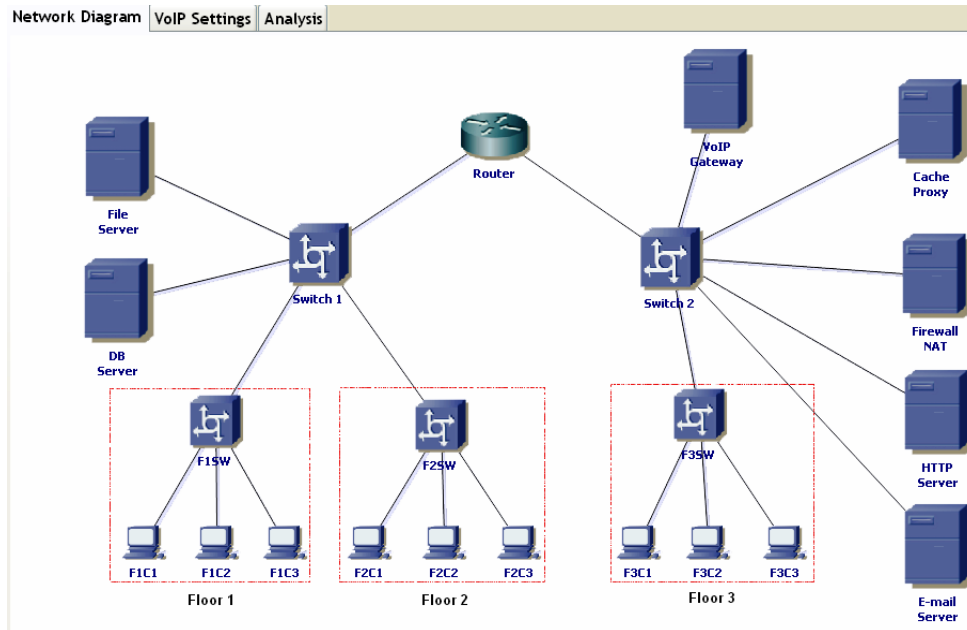
Figure 5. Corresponding network diagram constructed by analytical simulator

Figure 6 shows the reports of throughput and delay analyses. Figure 6(a) reports the number of calls that can be supported based on bandwidth analysis. A total of 315 calls can be supported for the whole network. In order to identify possible bottlenecks, the report also shows individual calls that can be supported per node and per link. It is shown that the router is the bottleneck, and supporting calls more than 315 calls would require definitely a replacement for the router. Figure 6(b) reports the number of calls that can be supported based on network analysis. A total of 313 calls can be supported such that the network delay of any of the specified VoIP flows does not exceed the required 80 ms. The figure shows that with 313 calls, a network delay of 16.76 ms will be introduced. This means when adding one more call, the network delay of a maximum of 80 ms was exceeded. The report of Figure 6(b) also exhibits the network delay per flow or path. In our example, there were a total of nine VoIP flows. As shown, the first triple is for intra-floor flows. The second triple is for inter-floor flows. And the third triple is for external flows. Such information gives insight on the source of the delays as well as the path that causing most of the delays. As the figure shows, the inter-floor flows of `F1C1-F3C2` and `F2C1-F3C2` experience the largest delays, as they pass through the router.

Therefore, it can be concluded that the network can support up to 313 calls with end-to-end delay of 16.76 ms. The network delay (for this particular network topology) is the more dominant factor (than throughput) in determining the number of voice calls to be supported. This is consistent with results obtained using MATLAB-based analysis and OPNET-based simulation in [11,12].
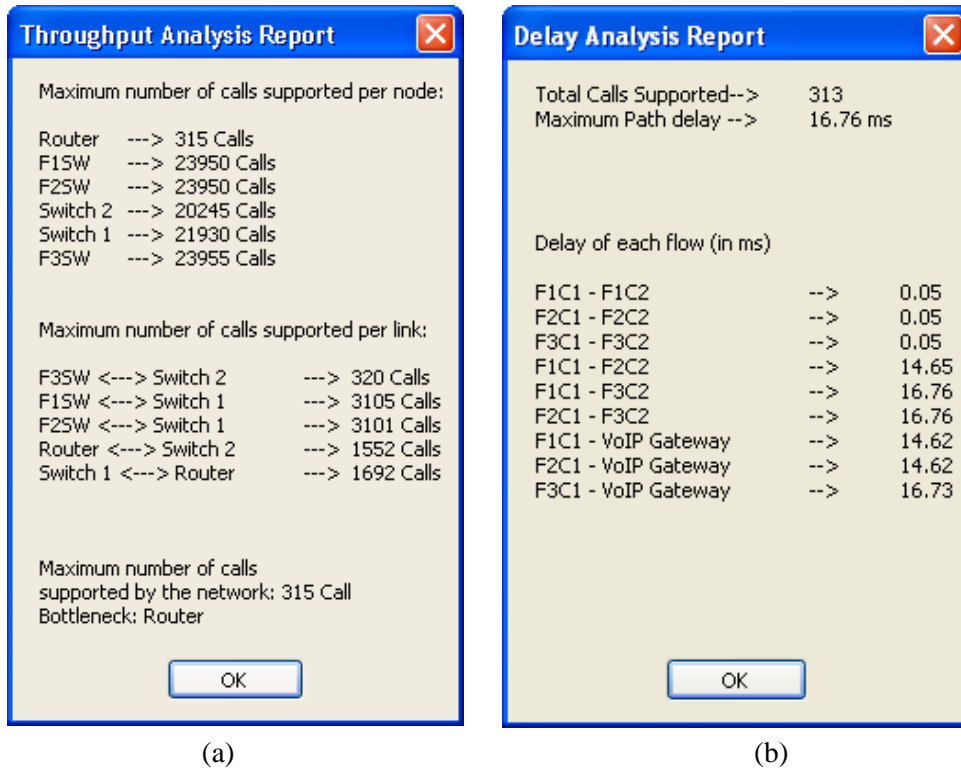
Throughput Analysis Report

Maximum number of calls supported per node:

Router   ---> 315 Calls
F1SW   ---> 23950 Calls
F2SW   ---> 23950 Calls
Switch 2 ---> 20245 Calls
Switch 1 ---> 21930 Calls
F3SW   ---> 23955 Calls

Maximum number of calls supported per link:

F3SW <---> Switch 2   ---> 320 Calls
F1SW <---> Switch 1   ---> 3105 Calls
F2SW <---> Switch 1   ---> 3101 Calls
Router <---> Switch 2   ---> 1552 Calls
Switch 1 <---> Router   ---> 1692 Calls

Maximum number of calls
supported by the network: 315 Call
Bottleneck: Router

OK

Delay Analysis Report

Total Calls Supported-->   313
Maximum Path delay -->   16.76 ms

Delay of each flow (in ms)

F1C1 - F1C2   -->   0.05
F2C1 - F2C2   -->   0.05
F3C1 - F3C2   -->   0.05
F1C1 - F2C2   -->   14.65
F1C1 - F3C2   -->   16.76
F2C1 - F3C2   -->   16.76
F1C1 - VoIP Gateway -->   14.62
F2C1 - VoIP Gateway -->   14.62
F3C1 - VoIP Gateway -->   16.73

OK

(a)   (b)

Figure 6. Throughput Analysis Report (a) and Delay Analysis Report (b)

4   Concluding Remarks

The paper presented a GUI-based VoIP analytical simulator with drag-and-drop features to construct a generic network topology. The simulator makes it easy for network designers and planners to assess network readiness and support for VoIP. Prior to the purchase and deployment of VoIP equipment, it is possible to predict the number of VoIP calls that can be sustained by a given network while satisfying QoS requirements of existing network services and leaving adequate capacity for future growth. The results obtained by the simulator are consistent with results of previous work reported in [11,12]. The results of [11,12] were based on results obtained by Matlab and OPNET simulation. In this paper, only peer-to-peer voice calls were considered. As a future work, one can consider expanding the simulator's features to implement important VoIP options such as VoIP conferencing and messaging, as well as other important multimedia and network services such as videoconferencing.

One related issue to our cased study and work presented in this paper is whether our simulator will be able to determine the number of VoIP calls that can be supported for an existing VPN (Virtual Private Network), i.e. a private data network with multiple branches or campuses connected via the Internet or private leased lines, which are typically owned by a third-party network service provider. Our analysis and simulation would also

be applicable in this situation. To account for the external network, the network engineer has to know the worst-case end-to-end delay and bandwidth (which are usually agreed on and guaranteed by the network service provider) between the border points connecting the branches. As a verification and alternative option, the engineer can perform measurement at these border points to determine the actual provided delay and available bandwidth. Such an external network can then be *approximately* modeled as a link with a given bandwidth and delay, or a as a router with a given processing rate and delay. The model for the link is appropriate for a leased line, whereas the router is more appropriate for a data network or the Internet.

Acknowledgements

References

[1] M. Bearden, L. Denby, B. Karacali, J. Meloche, and D. T. Stott, "Assessing Network Readiness for IP Telephony," Proceedings of IEEE International Conference on Communications, ICC02, vol.4, 2002, pp. 2568-2572

[2] B. Karacali, L. Denby, and J. Melche, "Scalable Network Assessment for IP Telephony," Proceedings of IEEE International Conference on Communications (ICC04), Paris, June 2004, pp. 1505-1511.

[3] Eurescom, H.323 Studies, http://www.eurescom.de/~public-webspace/P800-series/P807/start/printerversion.asp

[4] NetIQ, Vivinet Assessor, http://www.netiq.com

[5] BMCsoftware, "PATROL DashBoard," http://www.bmc.com

[6] Radvision, H.323 Protocol Toolkit, http://www.radvision.com

[7] Omegon, "White Paper: NetAlly," http://www.omegon.com/

[8] Lucent Technologies, "VitalSuite," http://www.qip.lucent.com

[9] VideNet Scout Resources, http://scout.video.unc.edu

[10] P. Calyam, W. Mandrawa, M. Sridharan, A. Khan, P. Schopis, "H.323 Beacon: An H.323 application related end-to-end performance troubleshooting tool", ACM SIGCOMM 2004 Workshop on Network Troubleshooting (NetTs), 2004.

[11] K. Salah and A. Alkhoraidly, "An OPNET-based Simulation Approach for Deploying VoIP," International Journal of Network Management, John Wiley, Vol. 16(3-4) (2006, pp. 159-183.

[12] K. Salah, "On the Deployment of VoIP in Ethernet Networks: Methodology and Case Study," International Journal of Computer Communications, Elsevier Science, Vol. 29(8) (2006), pp. 1039-1054.

[13] Salah, K., and Almashari, M., "An Analytical Tool to Assess Readiness of Existing Networks for Deploying IP Telephony", Proceeding of 11th IEEE Symposium on Computers and Communications, IEEE ISCC 2006, Sardinia, Italy, June 27-30, 2006, pp. 302-305

[14] Recommendation H.323, "Packet-based Multimedia Communication Systems," ITU, 1997.

[15] Recommendation G.114, "One-Way Transmission Time," ITU, 1996.

[16] ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality," www.itu.in/publications/main_publ/itut.html

[17] R. Ahuja, T. Magnanti, and J. Orlin. Network Flows. Prentice Hall, Englewood Cliffs NJ, 1993.

[18] Cisco Systems Inc., "Cisco 2621 Modular Access Router Security Policy," 2001, http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/secure/2621rect.pdf

[19] 3Com, "3Com Networking Product Guide," April 2004, http://www.3com.co.kr/products/pdf/productguide.pdf